

NNTP ActiveX Control

**NNTP Services
ActiveX Control
for Microsoft® Windows™**

Version 5.2

**Copyright © 1995 - 2003 by Distinct Corporation
All rights reserved**

Table of Contents

Table of Contents	2
1 Overview	5
1.1 Introduction	5
1.2 Usage	5
1.3 Property Summary	7
1.4 Event Summary	8
1.5 Method Summary	9
1.6 D_NNTP.TXT	10
2 Properties	13
2.1 AccessType	13
2.2 Action	14
2.3 ArticleId	17
2.4 Date	18
2.5 DirectBufSize	19
2.6 DirectCmd	20
2.7 EndArticleNo	21
2.8 FirewallPort	22
2.9 FirewallServer	23
2.10 FwAddrType	24
2.11 FwAuthMethods	25
2.12 FwPassword	26
2.13 FwSocksVer	27
2.14 FwUsername	28
2.15 Group	29
2.16 Host	30
2.17 NewCmdLine	31
2.18 NewType	32
2.19 Position	33
2.20 Reply	34
2.21 ReplyCode	35
2.22 Result	36
2.23 SendData	37
2.24 StartArticleNo	38
2.25 Timeout	39
2.26 UseProperty	40
2.27 UseVariant	41
3 Events	42
3.1 OnClose	42
3.2 OnConnect	43
3.3 OnDirect	44
3.4 OnError	45
3.5 OnForward	47
3.6 OnForwardB	48
3.7 OnHelp	50
3.8 OnList	51
3.9 OnNew	52
3.10 OnOverview	53
3.11 OnPost	54
3.12 OnPostB	56
3.13 OnRetrieve	57

4 Methods	59
4.1 Abort	59
4.2 AsyncConnect	60
4.3 CancelAsyncConnect	61
4.4 Connect	62
4.5 Direct	63
4.6 Disconnect	64
4.7 Forward	65
4.8 ForwardB	66
4.9 FwConnect	67
4.10 Help	69
4.11 List	70
4.12 NewArticles	71
4.13 NewGroups	72
4.14 Next	73
4.15 Overview	74
4.16 Post	75
4.17 PostB	76
4.18 PostBuf	77
4.19 Previous	78
4.20 RetrieveAll	79
4.21 RetrieveBody	80
4.22 RetrieveHead	81
4.23 RetrieveStat	82
4.24 Set	83
4.25 Slave	84

1 Overview

1.1 Introduction

The News Network Transfer Protocol (NNTP) is used to build applications that access Usenet, a public discussion forum made up of various news groups. The Distinct NNTP client ActiveX control allows you to read, post or list news items in your application. You will be able to navigate through all the available network news groups on a server and to retrieve or post news articles. A customized application could automatically sort news articles into categories based on the subject to save the user time scanning through many irrelevant articles. SOCKS 5 and SOCKS 4 support is embedded so that users can connect from the local machine to the remote server through a firewall proxy.

1.2 Usage

See the section entitled "Using Distinct ActiveX controls in various environments" on how to add the control to your project.

After placing an NNTP ActiveX control into a form, some properties can be set at design time. The Host property is usually set at run time right before a session is established. This allows the application to request the host name or internet address of the NNTP server from the user. If an application will always connect to the same host, then the Host property can also be set at design time to minimize user interaction.

Some properties (such as Action and Reply properties) can only be accessed at run time. To establish an NNTP session, the value of the Action property must be set to ACTION_CONNECT (or the Connect method called) or ACTION_ASYNC_CONNECT (or the AsyncConnect method called). The Host property is required to establish the connection. While ACTION_CONNECT (or the Connect method) blocks until the connection is established the ACTION_ASYNC_CONNECT (or the AsyncConnect method) returns immediately. If the connection can be established, the OnConnect event will occur.

The NNTP ActiveX control supports the Username/Password authentication protocol. If the application chooses to authenticate the user before actually establishing a connection with the server it must set the FwAuthMethods property to "2" . The FwUsername and FwPassword properties must be set to a valid user id and password respectively.

Since the NNTP ActiveX control supports both Socks version 5 and version 4 the application is given an option to specify the Socks version in the FwSocksVer property. If the application is not sure about the server version then they can leave the property empty and the NNTP ActiveX control will automatically determine the server version.

For some users, the local machine is located on a different subnet than the remote host and the only way of communication is through a firewall. If this is the case, then the built-in firewall support of the NNTP ActiveX control can be used to establish an NNTP session via a firewall by setting the Action property to ACTION_FW_CONNECT (or by calling the FwConnect method). In addition to setting the properties for a regular NNTP session, the FirewallServer and FirewallPort properties must also be set before setting the Action property to ACTION_FW_CONNECT. If the connection can be established, the OnConnect event will occur before the next line of code is reached.

To make a news group current, set the Group property to the news group of interest. A complete list of all news groups supported by the server can be obtained by setting the value of the Action property to ACTION_LIST (or by calling the List method). Within the current news group, articles or portions of articles can be retrieved when the Action property is set to ACTION_RETRIEVE (or when the Retrieve method is used).

An application can also post news articles by setting the Action property to ACTION_POST (or by calling the Post method). The application must define the header and the body of the article in one or more OnPost events. To get help information about available commands on the server, the Action property must be set to ACTION_HELP (or call the Help method). To get an overview of a specified range of news articles, the Action property must be set to ACTION_OVERVIEW (or by call the Overview method).

A command can be sent directly to the NNTP server by setting the DirectCmd property to a NNTP command (or by calling the Direct method). The OnDirect event will deliver the servers reply, in case of error the OnError event will be fired.

The value of the Result property indicates the success or failure of the last command executed on the NNTP server. This property can be read at any time and may be checked after initiating an operation.

Once a connection is no longer needed, the Action property must be set to ACTION_DISCONNECT (or the Disconnect method must be called). After the session is disconnected, the OnClose event will occur before the next line of code is reached. Applications must disconnect all connected sessions before terminating.

1.3 Property Summary

AccessType

Type of access to retrieve article

Action

Connect or disconnect an NNTP session, read news, post articles, abort session

ArticleId

Unique id or number of article

Date

Current time and date

DirectBufSize

Length of the reply buffer for the DirectCmd or Direct method

DirectCmd

Command to send to server

EndArticleNo

Last article in the set of articles to overview

FirewallPort

Firewall server port

FirewallServer

Name or dotted decimal internet address of firewall server

FwAddrType

The address type of the destination host

FwAuthMethods

The authentication method

FwPassword

A valid password

FwSocksVer

The Socks server version

FwUsername

A valid username

Group

Current news group

Host

Name of host or dotted decimal internet address

NewCmdLine

Command line arguments for new groups or new news articles

NewType

New groups or new news articles

Position

Current news article in a selected news group

Reply

Last response sent by the NNTP server

ReplyCode

Integer equivalent of the reply sent by the NNTP server

Result

Result of last command executed by NNTP server

SendData

Send data buffer

StartArticleNo

First article in the set of articles to overview

Timeout

Reply timeout

UseProperty

Send data by parameter or property

UseVariant

Forward and post articles as binary or ascii data

1.4 Event Summary

OnClose

NNTP connection has been closed

OnConnect

NNTP connection has been established

OnDirect

Reply from server has been received

OnError

Local error has occurred

OnForward

More data of article to be forwarded can be sent

OnForwardB

More data of article to be forwarded can be sent as binary

OnHelp

More data containing help information has been received

OnList

More data containing list of news groups has been received

OnNew

More data containing new group or new article information has been received

OnOverview

More data of article summary has been received

OnPost

More data of article to be posted can be sent

OnPostB

More data of article to be posted can be sent as binary

OnRetrieve

More data of article to be retrieved has been received

1.5 Method Summary

Abort

Abort current action

AsyncConnect

Initiate an asynchronous connect to the NNTP server

CancelAsyncConnect

Cancel an asynchronous connect to the NNTP server

Connect

Connect to NNTP server

Direct

Send a command to the NNTP server

Disconnect

Close connection to NNTP server

Forward

Forward an article

ForwardB

Forward an article with binary data

FwConnect

Establish connection via a firewall

Help

Show help information

List

List all available news groups

NewArticles

Get new news articles

NewGroups

Get new news groups

Next

Go to next article

Overview

Retrieve an overview of a specified range of news articles

Post

Post an article to a news group

PostB

Post an article to a news group with binary data

PostBuf

Post an article to a news group using buffers

Previous

Go to previous article

RetrieveAll

Retrieve an entire article

RetrieveBody

Retrieve the message portion of an article

RetrieveHead

Retrieve the article header

RetrieveStat

Retrieve the article statistics

Set

Go to a specified article

Slave

NNTP server becomes slave server

1.6 D_NNTP.TXT

The following provides a complete listing of the D_NNTP.TXT definition file. If your application uses more than one Distinct ActiveX control in the same form, then some definitions will conflict. For example, the FTP Client ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 3
```

in the D_FTP.TXT file and the Telnet ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 2
```

in the D_TNET.TXT file. To avoid this conflict, you must rename at least one of the constants (for example, FTP_ACTION_DISCONNECT or TNET_ACTION_DISCONNECT).

```
' NNTP ActiveX Control
' (C) Copyright 1995 - 1998 by Distinct Corporation
' All rights reserved

' article retrieval options
Global Const ARTICLE_ALL = 1
Global Const ARTICLE_BODY = 2
Global Const ARTICLE_HEAD = 3
Global Const ARTICLE_STAT = 4

' article position options
Global Const POSITION_PREV = 1
Global Const POSITION_NEXT = 2
Global Const POSITION_SET = 3

' new item options
Global Const NEW_GROUPS = 1
Global Const NEW_NEWS = 2
```

```
' actions
Global Const ACTION_NONE = 0
Global Const ACTION_CONNECT = 1
Global Const ACTION_DISCONNECT = 2
Global Const ACTION_LIST = 3
Global Const ACTION_POST = 4
Global Const ACTION_FORWARD = 5
Global Const ACTION_NEW = 6
Global Const ACTION_POSITION = 7
Global Const ACTION_RETRIEVE = 8
Global Const ACTION_HELP = 9
Global Const ACTION_SLAVE = 10
Global Const ACTION_ABORT = 11
Global Const ACTION_FW_CONNECT = 12
Global Const ACTION_OVERVIEW = 13
Global Const ACTION_ASYNC_CONNECT = 14
Global Const ACTION_CANCEL_ASYNC_CONNECT = 15

' error codes
Global Const ERR_HOST_NOT_DEFINED = 1
Global Const ERR_NOT_CONNECTED = 2
Global Const ERR_IN_ACTION = 3
Global Const ERR_CANNOT_CONNECT = 4
Global Const ERR_NEW_TYPE_NOT_DEFINED = 5
Global Const ERR_INVALID_ARTICLE_ID = 6
Global Const ERR_POSITION_NOT_DEFINED = 7
Global Const ERR_ACCESS_TYPE_NOT_DEFINED = 8
Global Const ERR_GROUP_NOT_DEFINED = 9
Global Const ERR_INVALID_GROUP = 10
Global Const ERR_CANNOT_ABORT = 11
Global Const ERR_INVALID_TIMEOUT = 12
Global Const ERR_UNABLE_TO_LOAD = 13
Global Const ERR_FW_SERVER_NOT_DEFINED = 14
Global Const ERR_IN_TRANSFER = 15
Global Const ERR_CANNOT_CANCEL = 16
Global Const ERR_INSUFFICIENT_BUFSIZE = 17

' return values
Global Const NNTP_SUCCESS = 1001
Global Const NNTP_FAILURE = 1002
Global Const NNTP_CANNOT_INITIALIZE = 1003
Global Const NNTP_CANNOT_INIT_WINSOCK = 1004
Global Const NNTP_OUT_OF_MEMORY = 1005
Global Const NNTP_UNKNOWN_HOST = 1006
Global Const NNTP_CANNOT_ALLOC_SOCKET = 1007
Global Const NNTP_CANNOT_BIND_SOCKET = 1008
Global Const NNTP_HOST_NOT_RESPONDING = 1009
Global Const NNTP_CANNOT_SEND_COMMAND = 1010
Global Const NNTP_TIMED_OUT = 1011
Global Const NNTP_INVALID_HANDLE = 1012
Global Const NNTP_INVALID_OPTION = 1013
Global Const NNTP_IN_PROGRESS = 1014
Global Const NNTP_ABORTED = 1015
```


2 Properties

2.1 AccessType

Summary

Type of access to retrieve article.

Description

The AccessType property specifies the type of access when retrieving the article specified by the ArticleId property. The AccessType property must be set before the Action property is set to ACTION_RETRIEVE. The AccessType property can take one of the following values.

Value	Meaning
ARTICLE_ALL	Entire article
ARTICLE_BODY	Message portion of article
ARTICLE_HEAD	Article header
ARTICLE_STAT	Article statistics

If the value of Access type is set to ARTICLE_ALL, then the entire article is retrieved during ACTION_RETRIEVE. To retrieve only the header of the article, set the AccessType property to ARTICLE_HEAD. To retrieve only the message portion of the article, set the AccessType property to ARTICLE_BODY. The article will be delivered through OnRetrieve events. Setting the AccessType property to ARTICLE_STAT will cause the ACTION_RETRIEVE command to retrieve only the article statistics. The article statistics will **not** be delivered via OnRetrieve events. The application must access the Reply property to obtain the response from the server.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Nntp.ArticleId = "123456"  
Nntp.AccessType = ARTICLE_ALL      ' entire article  
Nntp.Action = ACTION_RETRIEVE
```

2.2 Action

Summary

Connect or disconnect an NNTP session, read news, post articles, abort session.

Description

The Action property controls the connection state of the NNTP ActiveX control. A session can be established or closed and a news article transfer can be initiated by assigning one of the following values to the property.

Value	Meaning
ACTION_CONNECT	Establish session.
ACTION_DISCONNECT	Close session.
ACTION_LIST	List all available news groups.
ACTION_NEW	List new groups or new articles.
ACTION_HELP	Helpful information.
ACTION_POSITION	Set position of article in current news group.
ACTION_RETRIEVE	Read a news article.
ACTION_POST	Post an article to a news group.
ACTION_FORWARD	Forward an article.
ACTION_SLAVE	NNTP server becomes slave server.
ACTION_ABORT	Abort current action.
ACTION_FW_CONNECT	Establish session via a firewall.
ACTION_ASYNC_CONNECT	Initiate an asynchronous connect
ACTION_CANCEL_ASYNC_CONNECT	Cancel an asynchronous connect

This property can be changed at run time only.

Before setting the Action property to ACTION_CONNECT or ACTION_ASYNC_CONNECT, the Host property must be initialized. The Host property must be set to the name or internet address (in dotted decimal notation) of the NNTP server. Most applications will prompt the user with a dialog box for the information to place into this property, but the Host property can also be specified at design time.

If the NNTP client machine is located on a different subnet than the remote NNTP server machine and the only form of communication between these two machines is through a firewall gateway, then the built-in firewall support of the NNTP ActiveX control can be used to established an NNTP session. To establish a connection with a remote NNTP Server through a firewall, set the Action property to ACTION_FW_CONNECT. Before setting the Action property to ACTION_FW_CONNECT, some properties must be initialized. The FirewallServer property must be set to the name or internet address (in dotted decimal notation) of the firewall server. The FirewallPort property must be set to the firewall service port. In addition to the FirewallServer and FirewallPort properties, the Host property must also be set as mentioned above. Depending on the type of address specified in the Host property the FwAddrType property must be accordingly set, if the Host property contains the IP address of the remote host then the FwAddrType must be set to FW_ADDR_IP4 and if contains a machine name then the FwAddrType property must contain FW_ADDR_DNS. Distinct NNTP ActiveX Control supports both SOCKS version 5 and SOCKS version 4, the application can specify the SOCKS version in the FwSocksVer property. If the SOCKS version is 5 then the application can specify a authentication method in the FwAuthMethods property, currently only the Username/Password (FwUsername and FwPassword) authentication protocol is supported.

Setting the Action property to ACTION_ASYNC_CONNECT initiates an asynchronous connect to the NNTP server. The control is returned to the application immediately without waiting for the

connection to be established. This action can be canceled at any time by setting the Action property to ACTION_CANCEL_ASYNC_CONNECT.

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. In the case of the Action property being set to ACTION_CONNECT or ACTION_FW_CONNECT then these events will occur before the next statement (i.e. the statement following the assignment of ACTION_CONNECT or ACTION_FW_CONNECT to the Action property) is executed. In the case of the Action property being set to ACTION_ASYNC_CONNECT then these events may occur later once the connection has been established (this may take several seconds). The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

The application can now select a current news group. To set the current news article in the selected news group, set the Action property to ACTION_POSITION after assigning an appropriate value to the Position property. Depending on the Position property, the current article is set to the next article, to the previous article or to the article number specified by the ArticleId property.

To read a news article, set the ArticleId to the unique id or number of the specific article. Then, set the Action property to ACTION_RETRIEVE. OnRetrieve events will deliver the news article. Depending on the AccessType property, the OnRetrieve events will deliver the entire article, the message portion of the article or the header portion of the article. If the AccessType property is set to ARTICLE_STAT, then the response from the server will contain only the article statistics. This response must then be obtained by accessing the Reply property.

To read news article summaries, set the StartArticleNo to the first article number and set the EndArticleNo to the last article of the set of article summaries to overview. Then, set the Action property to ACTION_OVERVIEW. OnOverview events will deliver all the news article summaries.

To obtain a list of all available news groups on the server, set the Action property to ACTION_LIST. The information will be delivered through OnList events. The Action property, if set to ACTION_HELP, provides the application with a list of all commands available on the NNTP server with an explanation of the command. The information will be delivered through OnHelp events. Before setting the Action property to ACTION_NEW, the NewType and the NewCmdLine properties must be set. The list of new news groups or article Ids will be retrieved from the server and delivered through OnNew events.

To post an article to a news group, the Action property must be set to ACTION_POST. OnPost events will occur to receive the article header and the body. The required headers, such as the date and the news group, must be specified. A news article can be forwarded to the current server by setting the Action property to ACTION_FORWARD. The ArticleId property should be set to the unique id of the article to identify the article to be forwarded before this action is invoked. OnForward events will occur to forward the article.

The Action property can be set to ACTION_SLAVE to inform the current NNTP server that the local machine is acting as an intermediate or "slave" NNTP server. This information is only sent to advise the NNTP server and may be ignored by the server. It is normally used only by NNTP servers.

Setting the Action property to ACTION_ABORT will abort any list, help, new, post, forward or retrieve action in progress. This is usually done during an OnList, OnHelp, OnNew, OnPost, OnForward or OnRetrieve event to ensure that no further event will occur. ACTION_ABORT resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. At

this point, the application must still close the connection by setting the Action property to ACTION_DISCONNECT.

Once a connection is no longer needed, the session can be terminated by setting the Action property to ACTION_DISCONNECT. An application must close all connections it has created before it quits.

The value of the Result property can be checked to determine if the action has been performed successfully.

There is no default value for this property.

Note

The AsyncConnect, CancelAsyncConnect, Connect, FwConnect, Disconnect, List, NewArticles, NewGroups, Help, Previous, Next, Set, RetrieveAll, RetrieveHead, RetrieveBody, RetrieveStat, Overview, Post, Forward, Slave and Abort methods accomplish the same as the above actions. Please check the reference pages of these methods for more detailed information on their usage.

Example

```
Nntp.Host = "speedy.distinct.com"  
Nntp.Action = ACTION_CONNECT
```

2.3 ArticleId

Summary

Unique id or number of article.

Description

The ArticleId property identifies the article to be retrieved, forwarded or to be made current. The ArticleId property can either be a unique message identifier that conforms to the RFC 822 format or an article number.

The ArticleId property must be set before setting the Action property to ACTION_FORWARD, ACTION_RETRIEVE or ACTION_POSITION. To retrieve an article, the ArticleId property can be either a unique id or a number. To forward an article, the ArticleId property can only be a unique id. To make an article current, the ArticleId property should point to a valid article number and the Position property must be set to POSITION_SET.

The unique id must have the format "*unique*"@"*full domain name*", where "*full domain name*" is the full name of the host at which the article entered the network, and "*unique*" should be an integer sequence number or a short string. The number must be converted to a string before it is assigned to the ArticleId property.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Nntp.ArticleId = "123456"  
Nntp.AccessType = ARTICLE_HEAD    'header'  
Nntp.Action = ACTION_RETRIEVE
```

2.4 Date

Summary

Current time and date.

Description

The Date property returns the current date and time. The format of the string returned by the Date property is compliant with the RFC 850 specification. This property can be accessed at any time.

The application can use the Date property to build the header for the article. The date field is one of the required fields in an NNTP header. In the OnPost event, the application can obtain the current date and time by accessing the Date property and then build the 'date' field for the header.

Please refer to RFC 850 - "Standard For Interchange of USENET Messages" for more information on the format of the date string and the required fields for the message header.

This property can be set at design time or changed at run time. There is no default value for this property.

Example

```
MsgBox Nntp.Date, 64, "Sample Program"
```

2.5 DirectBufSize

Summary

Set the size of the buffer used to get the reply from the server in the DirectCmd or Direct method.

Description

The Distinct NNTP ActiveX Control lets the application send command *as is* to the server. The response from the server is copied to a buffer and the OnDirect event is fired to deliver the data to the application. The default size of the reply buffer is 1K , if the response from the server is more than 1K then that extra data is truncated. To avoid this the application can set the size of the receiving buffer through the DirectBufSize property.

The size specified is not enough then the ActiveX control will fire an error (ERR_INSUFFICIENT_BUFSIZE) and the OnDirect event will be fired with the truncated data. The application at this point can read the DirectBufSize property to check the required size of the buffer. In order to get the entire data the application has to issue the command again.

This property can be set at run time or at design time, it should be set before calling the Direct method or setting the DirectCmd property to any command. The default value of this property is 1024.

Example

```
Nntp.DirectBufSize = 2048
Nntp.DirectCmd = "Help" + Chr(13) + Chr(10)      ' get list of help commands
If Nntp.Result = NNTP_SUCCESS Then
    MsgBox Nntp.Reply, 64, "Sample Program"
End If
```

2.6 DirectCmd

Summary

Command to send to NNTP server

Description

The DirectCmd property is used to send a command to the NNTP server. The command is sent *as is* to the NNTP server over the previously established connection.

The server's response to the NNTP command will be delivered to the application in an OnDirect event. In case of an error the OnError event will be fired. This action also sets the Reply, ReplyCode and Result properties. The value of the Result property should be checked to determine if the command was successful.

The application can set the size of the buffer that would receive the reply from the NNTP server in the DirectBufSize property. If the buffer is not large enough to hold the entire reply from the server the NNTP ActiveX control will fire an error (ERR_INSUFFICIENT_BUFSIZE) and the OnDirect method will be fired only with the truncated data. The application at this point can check the DirectBufSize property to find out the required length of the buffer.

This property can only be set at run time while a connection is established. There is no default value for this property.

Example

```
Nntp.DirectBufSize = 2048
Nntp.DirectCmd = "Help" + Chr(10)      ' get list of help commands
If Nntp.Result = NNTP_SUCCESS Then
    MsgBox Nntp.Reply, 64, "Sample Program"
End If
```

2.7 EndArticleNo

Summary

Last article in the set of articles to overview.

Description

The EndArticleNo property specifies the last article id in the set of articles to overview. The overview action retrieves the summaries of a set of articles starting with the one specified by the StartArticleNo property and ending with the one specified by the End ArticleNo property.

The EndArticleNo and StartArticleNo properties have to be set before setting the Action property to ACTION_OVERVIEW. If the action was successful, the summaries will be delivered through one or more OnOverview events.

This property can be set at design time or changed at run time. There is no default value for this property.

Example

```
Nntp.StartArticleNo = 1234  
Nntp.EndArticleNo = 1240  
Nntp.Action = ACTION_OVERVIEW
```

2.8 FirewallPort

Summary

Firewall server port.

Description

The FirewallPort property specifies the port on the firewall server through which a connection can be established. The FirewallPort property must be set before a connection is established with the remote NNTP server (by setting the Action property to ACTION_FW_CONNECT or by calling the FwConnect method).

This property can be changed at any time except after a connection has been established with the remote NNTP server (by setting the Action property to ACTION_CONNECT or ACTION_FW_CONNECT). The default value for this property is 1080.

Example

```
Nntp.FirewallServer = "127.43.101.10"  
Nntp.FirewallPort = 1080  
Nntp.Host = "127.43.101.12"  
Nntp.Action = ACTION_FW_CONNECT
```

2.9 FirewallServer

Summary

Name of firewall server or dotted decimal internet address.

Description

The FirewallServer property specifies the name or internet address of a firewall through which a connection is to be made. This property must be set before a connection can be established (by changing the Action property). There are three possible ways of specifying a firewall server name.

Machine Name

An application only needs to specify the name of the firewall server if the host is located on the same network as the local PC or if the internet address of the host is defined in the local hosts data base. If the firewall server is not on the local network, then the underlying protocol will route the traffic through a gateway. If the host is not defined in the local hosts database, then the underlying protocol will contact the name server to resolve the internet address of the firewall server.

Machine and Domain Name

An application needs to specify the machine name and the domain name if the host is not located on the same network as the local PC. Fully domain qualified machine names are written from left to right in ascending order (for example, *speedy.distinct.com*). If both machine and domain names are specified, then the underlying protocol will contact the name server to resolve the internet address of the firewall server.

Internet Address

Sometimes the user knows only the internet address of the firewall server that he or she wants to use. In this case, the internet address can be entered in what is known as the dotted decimal notation (for example, *127.43.101.12*). If the host identified by this address is not on the local network, then the underlying protocol will route the traffic through a gateway.

This property can be changed at design time and at run time before a connection has been established. There is no default value for this property.

Example

```
Nntp.FirewallServer = "127.43.101.10"  
Nntp.FirewallPort = 1080  
Nntp.Host = "127.43.101.12"  
Nntp.Action = ACTION_FW_CONNECT
```

2.10 FwAddrType

Summary

Address type of the destination host.

Description

The *FwAddrType* property specifies the address type in the *Host* property. This property must be set before a connection can be established (by changing the Action property to ACTION_FW_CONNECT or calling the method FwConnect). There are three possible ways of specifying a destination address in the *Host* Property and therefore the *FwAddrType* property can have three possible values:

FW_ADDR_IP4	Address is a version 4 IP address
FW_ADDR_DNS	Address is a DNS style domain name
FW_ADDR_IP6	Address is a version 6 IP address

The *FwAddrType* property must be set to FW_ADDR_IP4 if the application is specifying a Internet address in the dotted decimal notation (***198.211.122.133***) in the *Host* property. If the application wants to specify the machine name or machine name and domain name (for example ***speedy.distinct.com***) as the *Host* they must set the *FwAddrType* property to FW_ADDR_DNS.

Note that the FW_ADDR_IP6 is not supported currently.

This property can be changed at design time and at run time before a connection has been established.

The default value for this property is FW_ADDR_IP4.

Example

```
Nntp.FirewallServer = "sparky.distinct.com"
Nntp.FirewallPort = 1080
Nntp.FwAddrType = FW_ADDR_DNS
Nntp.Host = "speedy.distinct.com"
Nntp.Action = ACTION_FW_CONNECT
```

2.11 FwAuthMethods

Summary

Firewall authentication methods.

Description

The *FwAuthMethods* property specifies the authentication method that can be used when connecting to the firewall server. This property must be set before a connection can be established (by changing the Action property to ACTION_FW_CONNECT or calling the method FwConnect). The *FwAuthMethods* can be "0", "1" or "2" or a combination of "0", "1", "2", for example "01" or "12". "0" means that no authentication is required, "1" means GSSAPI and "2" means that a valid username and password is required. Currently only methods "0" and "2" are supported.

This property can be changed at design time and at run time before a connection has been established.

The default value for this property is "0".

Example

```
Nntp.FirewallServer = "sparky.distinct.com"  
Nntp.FirewallPort = 1080  
Nntp.FwAddrType = FW_ADDR_DNS  
Nntp.Host = "speedy.distinct.com"  
Nntp.FwAuthMethods = "2"  
Nntp.FwUsername = "joe"  
Nntp.FwPassword = "distinct"  
Nntp.Action = ACTION_FW_CONNECT
```

2.12 FwPassword

Summary

A valid password .

Description

The *FwPassword* property specifies a valid password that is required during authentication. A valid password is required when connecting to Socks version 5 server if the authentication method (*AuthMethod*) specified is "2" (Username/Password authentication protocol).

This property can be changed at design time and at run time before a connection has been established by setting the *Action* property to ACTION_FW_CONNECT or by calling the method FwConnect.

The property does not have any default value.

Example

```
Nntp.FirewallServer = "sparky.distinct.com"  
Nntp.FirewallPort = 1080  
Nntp.FwAddrType = FW_ADDR_DNS  
Nntp.Host = "speedy.distinct.com"  
Nntp.FwAuthMethods = "2"  
Nntp.FwUsername = "joe"  
Nntp.FwPassword = "distinct"  
Nntp.Action = ACTION_FW_CONNECT
```

2.13 FwSocksVer

Summary

The firewall server version.

Description

The *FwSocksVer* property is used specify the version of the Socks server. This property can have any one or a combination of the following values.

Value	Meaning
FW_VERSION5	The Socks version is 5.
FW_VERSION4	The Socks version is 4

If the application is not sure about the socks version then it can specify both FW_VERSION5 and FW_VERSION4. The Distinct firewall control will automatically detect the firewall server version and make the appropriate connection.

This property can be set at design time or at run time before a connection is established by calling the method FwConnect or setting the Action to ACTION_FW_CONNECT.

Example

```
Nntp.FirewallServer = "sparky.distinct.com"  
Nntp.FirewallPort = 1080  
Nntp.FwAddrType = FW_ADDR_DNS  
Nntp.Host = "speedy.distinct.com"  
Nntp.FwAuthMethods = "2"  
Nntp.FwUsername = "joe"  
Nntp.FwPassword = "distinct"  
Nntp.FwSocksVer = FW_VERSION5  
Firewall.Action = ACTION_FW_CONNECT
```

2.14 FwUsername

Summary

A valid user name.

Description

The *Username* property specifies a valid username. A valid user id is required when connecting to Socks version 5 server if the authentication method (*AuthMethods*) specified is "2" (Username/Password authentication protocol). It is mandatory when a connection needs to be established with Socks version 4 server.

This property can be changed at design time and at run time before a connection has been established by setting the Action property to ACTION_FW_CONNECT or by calling the method FwConnect.

The property does not have any default value.

Example

```
Nntp.FirewallServer = "sparky.distinct.com"  
Nntp.FirewallPort = 1080  
Nntp.FwAddrType = FW_ADDR_DNS  
Nntp.Host = "speedy.distinct.com"  
Nntp.FwAuthMethods = "2"  
Nntp.FwUsername = "joe"  
Nntp.FwPassword = "distinct"  
Nntp.Action = ACTION_FW_CONNECT
```

2.15 Group

Summary

Current news group.

Description

The Group property must be set to the news group that the user is interested in. There are many actions which can only be performed if a newsgroup is current, such as retrieving news articles, making an article current or getting new news articles.

The group name should be a valid group on the server. When the application sets the Group property, this group becomes the current news group if it specifies a valid group on the server. The response from the server should therefore be checked by either accessing the ReplyCode property or the Reply property to ensure that the group is valid.

To obtain a list of all available news groups, set the Action property to ACTION_LIST (or call the List method). The list of groups will be delivered through OnList events. The application can then set the Group property to one of the group names obtained using the ACTION_LIST function (or the List method).

The Reply property can be accessed to check the response from the server for error returns. The ReplyCode property can also be used by the application to check the severity of an error.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Nntp.Group = "misc.test"
```

2.16 Host

Summary

Name of host or dotted decimal internet address.

Description

The Host property specifies the name or internet address of an NNTP server. This property must be set before a session can be established. There are three possible ways of specifying an NNTP server.

Machine Name

An application only needs to specify the name of the NNTP server if the server is located on the same network as the local PC or if the internet address of the server is defined in the local host table. If the NNTP server is not on the local network, then the underlying protocol will route the traffic through a gateway. If the NNTP server is not defined in the local host table, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Machine and Domain Name

An application needs to specify the machine name and the domain name if the NNTP server is not located on the same network as the local PC. Fully domain qualified machine names are written from left to right in ascending order (for example, *speedy.distinct.com*). If both machine and domain names are specified, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Internet Address

Sometimes the user knows only the internet address of the NNTP server that he or she wants to use. In this case, the internet address can be entered in what is known as the dotted decimal notation (for example, *127.43.101.12*). If the NNTP server identified by this address is not on the local network, then the underlying protocol will route the traffic through a gateway.

This property can be changed at design time and at run time before a connection has been established. There is no default value for this property.

Example

```
Nntp.Host = "127.43.101.12"  
Nntp.Action = ACTION_CONNECT
```

2.17 NewCmdLine

Summary

Command line arguments for new groups or new news articles.

Description

The NewCmdLine property specifies the command line arguments for the ACTION_NEW function. Before setting the Action property to ACTION_NEW, the NewCmdLine and NewType properties must be set. The NewCmdLine property should contain the required command line parameters (date and time) and may contain optional command line parameters (GMT and distributions). The syntax is:

date time [GMT] [distributions].

The **date** parameter is required and must be given as 6 digits in the format YYMMDD, where YY specifies the last two digits of the year, MM the two digits of the month (with leading zero if less than 10) and DD the day of the month (with leading zero if less than 10).

The **time** parameter is also required and must be given as 6 digits in the format HHMMSS, where HH specifies the hours 00 through 23, MM the minutes 00 through 59 and SS the seconds 00 through 59. The time is assumed to be in the servers time zone unless the optional **GMT** parameter is specified.

For example, the NewCmdLine property could be set to "940305 000000". Setting the Action property to ACTION_NEW would then list the new groups or news articles created since March 5, 1994.

For more information on the command line parameters used to obtain new groups or new articles, please refer to RFC 977 - "Network News Transfer Protocol".

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Nntp.NewType = NEW_GROUPS  
Nntp.NewCmdLine = "940305 000000"  
Nntp.Action = ACTION_NEW
```

2.18 NewType

Summary

New groups or new news articles.

Description

The NewType property specifies whether new group information or new news article information should be obtained. This property must be set before setting the Action property to ACTION_NEW. The NewType property can take on one of the following values.

Value	Meaning
NEW_GROUPS	Get new news groups
NEW_NEWS	Get new news articles

If the value of NewType is NEW_GROUPS, then setting Action to ACTION_NEW will retrieve all the new groups created since the date and time specified by the NewCmdLine property. If the value of NewType is NEW_NEWS, then setting Action to ACTION_NEW will retrieve all the new articles in the current news group (specified by the Group property) created since the date and time specified by the NewCmdLine property. The list of groups or news articles will be delivered through one or more OnNew events.

For more information on the NewCmdLine property and the Group property, please refer to the sections which describe them.

The Reply property can be accessed to check the response from the server for error returns. The ReplyCode property can also be used by the application to check the severity of an error.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Nntp.NewType = NEW_GROUPS  
Nntp.NewCmdLine = "940305 000000"  
Nntp.Action = ACTION_NEW
```

2.19 Position

Summary

Current news article in a selected news group.

Description

The Position property specifies the current article in the current news group. The Position property can take on one of the following values.

Value	Meaning
POSITION_PREV	Go to previous article
POSITION_NEXT	Go to next article
POSITION_SET	Go to specified article

Before setting the value of the Action property to ACTION_POSITION, the Position property must be set. If the Position property is set to POSITION_PREV, then ACTION_POSITION will select the previous article as the current article. If the Position property is POSITION_NEXT, then ACTION_POSITION will select the next article as the current article. If the Position property is set to POSITION_SET, then the ArticleId property must specify a valid article number. In this case, ACTION_POSITION will select the article specified by the ArticleId property as the current article.

The Reply property can be accessed to check the response from the server for error returns. The ReplyCode property can also be used by the application to check the severity of an error.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Nntp.Position = POSITION_NEXT  
Nntp.Action = ACTION_POSITION
```

2.20 Reply

Summary

Last response sent by the NNTP server.

Description

The Reply property is used to access the last response that was received from the NNTP server. This property can only be accessed while a session is established. This property will contain the exact response from the NNTP server for the last command that was executed by the server.

If setting the Action property to ACTION_CONNECT, ACTION_FORWARD, ACTION_HELP, ACTION_LIST, ACTION_NEW, ACTION_POST, ACTION_RETRIEVE or ACTION_SLAVE (or if the equivalent method) returns an error, then the Reply property can be used to retrieve the error message sent by the host. This message will describe the reason for the failure of the action.

The Reply property can also be checked after setting the Group property to ensure that the specified news group is selected as the current group. This property will also be set if the DirectCmd property is set to a command (or the Direct method is called). In this case the Reply property will contain the reply from the server upon receiving the command, and not necessarily an error message.

The application can also check the value of the Result property to determine the exact nature of an error. For more details on error values, refer to the section explaining the Result property.

This property can only be read at run time while a connection is established. There is no default value for this property.

Example

```
Dim Message As String
```

```
Message = Nntp.ReplyCode & Nntp.Reply  
MsgBox Message, 64, "Sample Program"
```

2.21 ReplyCode

Summary

Integer equivalent of the reply sent by the NNTP server.

Description

The ReplyCode property is used to access the last response that was received from the NNTP server. This property can only be accessed while a session is established. This property will contain the integer equivalent of the response from the server for the last command that was executed by the NNTP server. The complete response string can be obtained by accessing the Reply property.

If setting the Action property to ACTION_CONNECT, ACTION_FORWARD, ACTION_HELP, ACTION_LIST, ACTION_NEW, ACTION_POST, ACTION_RETRIEVE or ACTION_SLAVE (or if the equivalent method) returns an error, then the ReplyCode property can be used to retrieve the error code sent by the host. This code will describe the reason for the failure of the action.

The ReplyCode property can also be checked after setting the Group property to ensure that the specified news group is selected as the current group. This property will also be set if the DirectCmd property is set to a command (or the Direct method is called). In this case the ReplyCode property will contain the reply code from the server upon receiving the command, this may not necessarily indicate an error.

The ReplyCode property provides more details on the exact nature of an error. For more information on the reply codes and their meaning, please refer to RFC 977 - "Network News Transfer Protocol".

The application can also check the value of the Result property to determine the exact nature of an error. For more details on error values, refer to the section explaining the Result property.

This property can only be read at run time while a connection is established. There is no default value for this property.

Example

```
Dim Message As String
```

```
Message = Nntp.ReplyCode & Nntp.Reply  
MsgBox Message, 64, "Sample Program"
```

2.22 Result

Summary

Result of last command executed by NNTP server.

Description

The Result property contains the result of the last operation by the NNTP server. This property can have any one of the following values.

Value	Meaning
NNTP_SUCCESS	Operation completed successfully.
NNTP_FAILURE	Host was unable to execute command.
NNTP_CANNOT_SEND_COMMAND	Underlying Windows Sockets kernel was unable to accept outgoing data.
NNTP_TIMED_OUT	Timeout occurred while waiting for response from host.
NNTP_UNKNOWN_HOST	Unable to resolve specified host name.
NNTP_HOST_NOT_RESPONDING	Unable to connect to NNTP port on specified host.
NNTP_CANNOT_INITIALIZE	Unable to initialize network drivers or support libraries.
NNTP_CANNOT_INIT_WINSOCK	Unable to initialize Windows Sockets transport library.
NNTP_OUT_OF_MEMORY	Not enough memory to allocate control structure.
NNTP_CANNOT_ALLOC_SOCKET	Windows Sockets was unable to allocate socket.
NNTP_CANNOT_BIND_SOCKET	Windows Sockets was unable to bind socket to local port.
NNTP_INVALID_HANDLE	Invalid connection handle was specified.
NNTP_INVALID_OPTION	Invalid connection option was specified
NNTP_IN_PROGRESS	Another operation is already in progress
NNTP_INSUFFICIENT_LENGTH	The buffer size to receive the reply from the server in response to the command passed through the Direct method or by setting the DirectCmd property is not large enough.
NNTP_ABORTED	Operation was aborted and connection has been closed.

The Result property reflects the result of the last operation, which was caused by setting the Action property or by calling a method.

The value of this property should be checked immediately after the operation. Accessing other properties may change the value of the property.

This property can be read at any time. There is no default value for this property.

Example

```
Nntp.ArticleId = "123456"
Nntp.AccessType = ARTICLE_STAT    'statistics'
Nntp.Action = ACTION_RETRIEVE
If Nntp.Result <> NNTP_SUCCESS Then
    MsgBox "Unable to retrieve article", 64, "Sample Program"
End If
```

2.23 SendData

Summary

Send data buffer

Description

The SendData property is used to send data in the OnPost and the OnForward events. If UseProperty is set to True, the control will get the data to send from the SendData property, otherwise it will get the data from the *Buffer* parameter. The UseProperty property should generally be set to False, and the *Buffer* event parameter used instead of this property. See the UseProperty for more information on when to use this property.

This property should be set in the OnPost or OnForward event. This property can be read at any time. There is no default value for this property.

Example

```

Sub Nntp_OnPost (Buffer As String, Length As Integer)
    Select Case dataIndex
        ' dataIndex (Integer) is initially 0
    Case 0:
        Nntp.SendData = "Date: " + Nntp.Date + Chr(10)      ' date
    Case 1:
        Nntp.SendData = "From: John" + Chr(10)              ' sender
    Case 2:
        Nntp.SendData = "NewsGroups: misc.test" + Chr(10)   ' news groups
    Case 3:
        Nntp.SendData = "Subject: Testing" + Chr(10)        ' subject
    Case 4:
        Nntp.SendData = Chr(10) + "This is a test"          ' data
    Case 5:
        Nntp.SendData = ""                                   ' end of data
    End Select
    dataIndex = dataIndex + 1
End Sub

Sub Nntp_OnForward (Buffer As String, Length As Integer)
    Select Case dataIndex
        ' dataIndex (Integer) is initially 0
    Case 0:
        Nntp.SendData = Chr(10) + Message                  ' data
    Case 1:
        Nntp.SendData = ""                                  ' end of data
    End Select
    dataIndex = dataIndex + 1
End Sub

```

2.24 StartArticleNo

Summary

First article in the set of articles to overview.

Description

The StartArticleNo property specifies the first article id in the set of articles to overview. The overview action retrieves the summaries of a set of articles starting with the one specified by the StartArticleNo property and ending with the one specified by the End ArticleNo property.

The StartArticleNo and EndArticleNo properties have to be set before setting the Action property to ACTION_OVERVIEW. If the action was successful, the summaries will be delivered through one or more OnOverview events.

This property can be set at design time or changed at run time. There is no default value for this property.

Example

```
Nntp.StartArticleNo = 1234  
Nntp.EndArticleNo = 1240  
Nntp.Action = ACTION_OVERVIEW
```

2.25 Timeout

Summary

Reply timeout.

Description

The Timeout property specifies the reply timeout period. The application can change the value of this property at design time or at run time.

The NNTP ActiveX control waits for a reply from the NNTP server before returning to the application. Because communication links or servers sometimes go down, the NNTP ActiveX control uses a default timeout of 20 seconds before returning the error value NNTP_TIMED_OUT if no response was received.

The application must specify the new timeout value in seconds. The value must be in the range of 1 to 999 seconds.

This property can be changed at design time or at run time. The default value for this property is 20.

Example

`Nntp.Timeout = 10`

2.26 UseProperty

Summary

Send data by parameter or property

Description

The UseProperty property is used to specify whether the *Buffer* parameter, or the SendData property should be assigned data in the OnPost and OnForward events. If UseProperty is set to True, the control will get the data to send from the SendData property, otherwise it will get the data from the Buffer parameter.

The UseProperty property is generally set to False, and the event parameters should be used in most cases. It should only be set to True in environments like Visual J++ where the control is unable to get the new value of the event parameter from the event. This is because Visual J++ has to use VBScript to catch and pass the ActiveX events, and VBScript is unable to pass back parameters to the control.

This property should be set before calling the setting the Action property to ACTION_POST or ACTION_FORWARD (or calling the Post or Forward methods). This property can be read at any time. The default value for this property is False.

Example

```
Nntp.UseProperty = False
```

2.27 UseVariant

Summary

Forward and post articles as binary or ascii data.

Description

The UseVariant property is used to specify whether data is to be forwarded or posted in binary or ascii form. If this property is set to True, the OnForwardB event will be fired to forward an article; otherwise, the OnForward event is fired. Similarly, the OnPostB event will be fired to post an article; otherwise, the OnPost event is fired.

This property should be set before forwarding or posting an article. This property can be read at any time. The default value for this property is False.

Example

```
Nntp.UseVariant = True
```

3 Events

3.1 OnClose

Summary

NNTP connection has been closed.

Description

The OnClose event occurs usually in response to the Disconnect or Abort methods (or in response to setting the Action property to ACTION_DISCONNECT or ACTION_ABORT). In some cases, the NNTP server may close a connection (for example because of a long period of inactivity). This will also trigger an OnClose event. In this case, the application must still call the Disconnect method (or set the Action property to ACTION_DISCONNECT) to free up all the resources allocated for the connection. However, this should not be done during the OnClose event because it might result in an infinite loop.

If this event occurs in response to the Disconnect method (or in response to setting the Action property to ACTION_DISCONNECT), it will occur before the next line of code is executed. However, if this event occurs in response to the Abort method (or in response to setting the Action property to ACTION_ABORT), the application must also close the connection by calling the Disconnect method (or by the setting the Action property to ACTION_DISCONNECT).

Normally, an application should simply set a flag in response to this event. Then, this flag can be checked directly after the Disconnect method is executed (the ACTION_DISCONNECT action) to make sure that the connection was actually terminated.

Example

```
Sub Nntp_OnClose ()
    If Connected = True Then
        Connected = False
        Nntp.Action = ACTION_DISCONNECT
    End If
End Sub
```

3.2 OnConnect

Summary

NNTP connection has been established.

Description

The OnConnect event occurs in response to the Connect, FwConnect or AsyncConnect method (or to setting the Action property to ACTION_CONNECT , ACTION_FW_CONNECT, or ACTION_ASYNC_CONNECT). When fired in response to calling the Connect or FwConnect method (or setting the Action property to ACTION_CONNECT or ACTION_FW_CONNECT) this event will occur before the next line of code is executed. When fired in response to the AsyncConnect method (or the Action property to ACTION_ASYNC_CONNECT) this event will occur after the connection has been established (this may take several seconds).

Normally, an application should simply set a flag in response to this event. Then, this flag can be checked after the Connect, FwConnect or AsyncConnect methods are executed (or the ACTION_CONNECT, ACTION_FW_CONNECT or ACTION_ASYNC_CONNECT) to verify whether the connection was actually established. If the connection could not be established, then the OnError event will be fired instead of the OnConnect event.

While handling the OnConnect event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent events.

Example

```
Sub Nntp_OnConnect ()  
    Connected = True  
End Sub
```

3.3 OnDirect

Summary

A reply to the direct command has been received from the server.

Description

The OnDirect event will occur in response calling the Direct method (or to setting the DirectCmd property). The OnDirect event delivers the reply from the server to the application.

While handling the OnDirect event, an application should not perform tasks, which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnDirect events.

Example

```
Sub Nntp_OnDirect (Text As String)
    List.Text = List.Text + Text
End Sub
```

3.4 OnError

Summary

Local error has occurred.

Description

The OnError event occurs when a property is accessed in an illegal way or when a connection with the NNTP server cannot be established. The table below lists all possible error codes delivered by this event.

Value	Meaning
ERR_HOST_NOT_DEFINED	Host name must be defined before connecting.
ERR_NOT_CONNECTED	Must be connected before accessing property.
ERR_IN_ACTION	Another action is in progress.
ERR_CANNOT_CONNECT	Unable to connect to remote host.
ERR_NEW_TYPE_NOT_DEFINED	New type not defined.
ERR_INVALID_ARTICLE_ID	Article name or number is invalid.
ERR_POSITION_NOT_DEFINED	Sender name not defined.
ERR_ACCESS_TYPE_NOT_DEFINED	Access type not defined.
ERR_GROUP_NOT_DEFINED	Group name not defined.
ERR_INVALID_GROUP	Invalid group name specified.
ERR_CANNOT_ABORT	Cannot abort action.
ERR_INVALID_TIMEOUT	Invalid timeout value specified.
ERR_FW_SERVER_NOT_DEFINED	Firewall server is not specified.
ERR_IN_TRANSFER	UseVariant property cannot be set while forward or post is in progress.
ERR_CANNOT_CANCEL	Cannot cancel asynchronous connect.
ERR_INSUFFICIENT_BUFSIZE	The buffer size to receive the reply from the server in response to the command passed through the Direct method or by setting the DirectCmd property is not large enough.

The following describes each error in more detail.

ERR_HOST_NOT_DEFINED

The Host property was not set before attempting to establish a connection.

ERR_NOT_CONNECTED

An attempt was made to access the Reply property or to set the Action property to a value other than ACTION_CONNECT (or the Connect method) without being connected.

ERR_IN_ACTION

Another action is already in progress.

ERR_CANNOT_CONNECT

NNTP server is unreachable. The Host property may be set incorrectly or the host may be down.

ERR_NEW_TYPE_NOT_DEFINED

The NewType property must be set to either NEW_GROUPS (NewGroups method) or NEW_NEWS (NewArticles method) before the Action property can be set to ACTION_NEW.

ERR_INVALID_ARTICLE_ID

The ArticleId property must contain a unique id or a valid number before the Action property can be set to ACTION_RETRIEVE (RetrieveAll, RetrieveHead, RetrieveBody and RetrieveStat methods). For ACTION_POSITION (Previous, Next and Set methods), the ArticleId property must

be a valid number and for ACTION_FORWARD (Forward method), the ArticleId property must be a unique id.

ERR_POSITION_NOT_DEFINED

The Position property must be set before the Action property can be set to ACTION_POSITION.

ERR_ACCESS_TYPE_NOT_DEFINED

The AccessType property must be set to the type of article access before retrieving an article.

ERR_GROUP_NOT_DEFINED

There is no current news group selected. The Group property must be set to a valid group name.

ERR_INVALID_GROUP

The Group property contains an invalid group name.

ERR_CANNOT_ABORT

The Action property can be set to ACTION_ABORT (or the Abort method can be called) only during ACTION_LIST (List method), ACTION_NEW (NewArticles and NewGroups methods), ACTION_HELP (Help method), ACTION_POST (Post method), ACTION_RETRIEVE (RetrieveAll, RetrieveHead, RetrieveBody and RetrieveStat methods) or ACTION_FORWARD (Forward method). ACTION_ABORT (or the Abort method) cannot be used in any other circumstances.

ERR_INVALID_TIMEOUT

The Timeout property contains an invalid timeout specification. The range for the timeout property is between 1 and 999.

ERR_FW_SERVER_NOT_DEFINED

Firewall server has not been initialized. Connections to remote host via a firewall can only be established if the firewall server is defined. Make sure that the FirewallServer property has been set to a valid firewall server name or IP address.

ERR_IN_TRANSFER

An attempt has been to change the UseVariant property while forward or post article is in progress.

ERR_CANNOT_CANCEL

The Action property can be set to ACTION_CANCEL_ASYNC_CONNECT (or the CancelAsyncConnect method called) only during ACTION_ASYNC_CONNECT (AsyncConnect method). ACTION_CANCEL_ASYNC_CONNECT (or the CancelAsyncConnect method) cannot be used in any other circumstances.

ERR_INSUFFICIENT_BUFSIZE

The buffer size to receive the reply from the server in response to the command passed through the Direct method or by setting the DirectCmd property is not large enough. The application can increase the size of the buffer through the DirectBufSize property.

Example

```
Sub Nntp_OnError (ErrorCode As Integer)
    If ErrorCode = ERR_CANNOT_CONNECT Then
        MsgBox "Unable to connect to remote host", 64, "Sample Program"
    End If
End Sub
```

3.5 OnForward

Summary

More data of article to be forwarded can be sent.

Description

The OnForward event occurs in response to the Action property being set to ACTION_FORWARD (or in response to the Forward method). The event accepts the data from the application and delivers it to the NNTP server. Normally, only an NNTP server forwards articles to other NNTP servers.

The application must send the header and the body of the article in one or more OnForward events. The RFC 850 - "Standard for Interchange of USENET Messages" describes the specifications for the header and body of the messages.

The *Length* argument will indicate the maximum number of bytes that can be assigned to *Buffer*. The application must assign all or a portion of the article (up to the maximum number of bytes specified by *Length*) to *Buffer*. To indicate the end of the message, the application should clear *Buffer* by assigning an empty string ("") to it. After returning an empty string buffer, no more OnForward events will occur. The application can also terminate the events by setting the Action property to ACTION_ABORT (or by calling the Abort method) during the OnForward event.

In some environments, such as Visual J++, the new value assigned to the *Buffer* parameter can not be successfully retrieved by the control. In these cases set UseProperty to True, and assign data to the SendData property instead of the *Buffer* parameter. The UseProperty property must be set to True before the Action property is set to ACTION_FORWARD (or before calling the Forward method) In this case treat the SendData property exactly as you would the *Buffer* parameter. If UseProperty is True the *Length* parameter will be ignored.

If the Action property is set to ACTION_ABORT (or if the Abort method) is used during this event, then the forward action in progress will be canceled and no further such events will occur. ACTION_ABORT (or Abort method) resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by setting the Action property to ACTION_DISCONNECT (or by calling the Disconnect method).

While handling the OnForward event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnForward events.

Example

```
Sub Nntp_OnForward (Buffer As String, Length As Integer)
    Select Case dataIndex
        ' dataIndex (Integer) is initially 0
        Case 0:
            Buffer = Chr(10) + Message           ' data
        Case 1:
            Buffer = ""                          ' end of data
    End Select
    dataIndex = dataIndex + 1
End Sub
```

3.6 OnForwardB

Summary

More data of article to be forwarded can be sent with binary.

Description

The OnForwardB event occurs in response to the Action property being set to ACTION_FORWARD (or in response to the Forward method) with the UseVariant property set to True. If the UseVariant property is False, then the OnForward event will be fired instead. The OnForwardB event does not accept data from the application, but expects the application to call the ForwardB method with the data to send. The Byte argument passed to this event indicates the maximum number of bytes of data that can be sent at that moment. Normally, only an NNTP server forwards articles to other NNTP servers.

The application must send the header and the body of the article in one or more OnForwardB events. The RFC 850 - "Standard for Interchange of USENET Messages" describes the specifications for the header and body of the messages.

In the OnForwardB event, the application should call the ForwardB method and pass the necessary parameters to the method. The Buffer parameter should contain the data to be sent and the Bytes parameter is the number of bytes that is passed as the Buffer parameter. To indicate the end of the message, the application should pass 0 as the Bytes parameter. After this, no more OnForwardB events will occur. The application can also terminate the events by setting the Action property to ACTION_ABORT (or by calling the Abort method) during the OnForward event.

If the Action property is set to ACTION_ABORT (or if the Abort method) is used during this event, then the forward action in progress will be canceled and no further such events will occur. ACTION_ABORT (or Abort method) resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by setting the Action property to ACTION_DISCONNECT (or by calling the Disconnect method).

While handling the OnForwardB event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnForwardB events.

Example

```
Sub Nntp_OnForwardB (Bytes As Long)
    Dim Buffer(1 to Bytes) As Byte
    Dim i As Integer

    If (Siz > Bytes) Then
        For i = 1 to Bytes
            Get #1, , Buffer(i)
        Next i
        Result = Nntp.ForwardB(Buffer, Bytes)           ' send message
        Siz = Siz - Bytes
    ElseIf (Siz > 0) Then
        For i = 1 to Siz
            Get #1, , Buffer(i)
        Next i
        Result = Nntp.ForwardB(Buffer, Siz)           ' send message
        Siz = 0
    Else
        Result = Nntp.ForwardB(Buffer, 0)             ' end of message
```

End If
End Sub

3.7 OnHelp

Summary

More data containing help information has been received.

Description

The OnHelp event will occur in response to the Action property being set to ACTION_HELP (or in response to the Help method). The Action property can be set to ACTION_HELP (or the Help method can be used) only when a connection has been established.

One or more OnHelp events will occur to deliver helpful information to the application about all available commands on the NNTP server.

If the Action property is set to ACTION_ABORT (or if the Abort method is used) during this event, then the help action in progress will be canceled and no further such events will occur. ACTION_ABORT (or Abort method) resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by setting the Action property to ACTION_DISCONNECT (or by calling the Disconnect method can also be used).

While handling the OnHelp event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnHelp events.

Example

```
Sub Nntp_OnHelp (Text As String)
    List.Text = List.Text + Text
End Sub
```

3.8 OnList

Summary

More data containing list of news groups has been received.

Description

The OnList event will occur in response to the List method (or the Action property being set to ACTION_LIST) after a connection has been established.

One or more OnList events will occur to deliver the list of all available news groups on the NNTP server.

If the Abort method is used (or the Action property is set to ACTION_ABORT) during this event, then the list action in progress will be canceled and no further such events will occur. The Abort method (or the ACTION_ABORT property) resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must close the connection by calling the Disconnect method (or setting the Action property to ACTION_DISCONNECT).

While handling the OnList event, an application should not perform tasks, which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnList events.

Example

```
Sub Nntp_OnList (Text As String)
    List.Text = List.Text + Text
End Sub
```

3.9 OnNew

Summary

More data containing new group or new article information has been received.

Description

The OnNew event occurs in response to the NewArticles or NewGroups method (or to the Action property being set to ACTION_NEW, with the NewType property set). When fired in response to the NewGroups method, this event delivers a portion or all of the list of news groups. When fired in response to the NewArticles method it delivers the news articleId numbers in the current news group that were created after the date specified by the NewArticles method.

One or more OnNew events will occur to deliver the list of new groups or new articles in the current news group.

If the Abort method is used (or the Action property is set to ACTION_ABORT) during this event, then the list action in progress will be canceled and no further such events will occur. The Abort method (or ACTION_ABORT property) resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by calling the Disconnect method (or by setting the Action property to ACTION_DISCONNECT)

While handling the OnNew event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnNew events.

Example

```
Sub Nntp_OnNew (Text As String)
    List.Text = List.Text + Text
End Sub
```

3.10 OnOverview

Summary

More data of article summary has been received.

Description

The OnOverview event occurs in response the Overview method being called (or to the Action property being set to ACTION_OVERVIEW with the StartArticleNo and EndArticleNo properties set first). The event delivers the news article summaries from the NNTP server. The event delivers all the news article summaries starting with the one specified by the StartArticleNo property and ending with the one specified by the EndArticleNo property.

If the Abort method is called (or the Action property is set to ACTION_ABORT) during this event, then the overview action in progress will be cancelled and no further such events will occur. Aborting resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by calling the Disconnect method (or by setting the Action property to ACTION_DISCONNECT).

While handling the OnOverview event, an application should not perform tasks which have the potential of requiring a lot of time to complete such as generating a message box. A substantial delay could cause the application to not receive subsequent OnOverview events.

Example

```
Sub Nntp_Overview (Text As String)
    List.Text = List.Text + Text
End Sub
```

3.11 OnPost

Summary

More data of article to be posted can be sent.

Description

The OnPost event occurs in response to the Post method (or in response to the Action property being set to ACTION_POST). The event accepts message data from the application and delivers it to the NNTP server.

The application must send the header and the body of the article in one or more OnPostB events. The RFC 850 - "Standard for Interchange of USENET Messages" describes the specifications for the header and body of the messages. There are some required headers, such as date, news group and sender. The current date and time can be obtained by accessing the Date property.

The *Length* argument will indicate the maximum number of bytes that can be assigned to *Buffer*. The application must assign all or a portion of the article (up to the maximum number of bytes specified by *Length*) to *Buffer*. To indicate the end of the message, the application should clear *Buffer* by assigning an empty string ("") to it. After returning an empty string buffer, no more OnPost events will occur.

In some environments, such as Visual J++, the new value assigned to the *Buffer* parameter can not be successfully retrieved by the control. In these cases set UseProperty to True, and assign data to the SendData property instead of the *Buffer* parameter. The UseProperty property must be set to True before the Action property is set to ACTION_POST (or before calling the Post method) In this case treat the SendData property exactly as you would the *Buffer* parameter. If UseProperty is True the *Length* parameter will be ignored.

If the Action property is set to ACTION_ABORT (or if the Abort method is used) during this event, then the post action in progress will be canceled and no further such events will occur. ACTION_ABORT(or Abort method) resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by setting the Action property to ACTION_DISCONNECT (or by calling the Disconnect method can also be used).

While handling the OnPost event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnPost events.

Example

```
Sub Nntp_OnPost (Buffer As String, Length As Integer)
  Select Case dataIndex
    ' dataIndex (Integer) is initially 0
    Case 0:
      Buffer = "Date: " + Nntp.Date + Chr(10)      ' date
    Case 1:
      Buffer = "From: John" + Chr(10)             ' sender
    Case 2:
      Buffer = "NewsGroups: misc.test" + Chr(10)  ' news groups
    Case 3:
      Buffer = "Subject: Testing" + Chr(10)       ' subject
    Case 4:
      Buffer = Chr(10) + "This is a test"         ' data
    Case 5:
      Buffer = ""                                  ' end of data
```

```
End Select  
DataIndex = DataIndex + 1  
End Sub
```

3.12 OnPostB

Summary

More data of article to be posted can be sent as binary.

Description

The OnPostB event occurs in response to the Post method (or to the Action property being set to ACTION_POST) with the UseVariant property set to True. If the UseVariant property is False, the OnPost event will occur instead. The OnPostB event does not accept data from the application, but expects the application to call the PostB method with the data to send. The Byte argument passed to this event indicates the maximum number of bytes of data that can be sent at that moment.

The application must send the header and the body of the article in one or more OnPostB events. The RFC 850 - "Standard for Interchange of USENET Messages" describes the specifications for the header and body of the messages. There are some required headers, such as date, news group and sender. The current date and time can be obtained by accessing the Date property.

if the Abort method is used (or if the Action property is set to ACTION_ABORT) during this event, then the post action in progress will be canceled and no further such events will occur. Aborting resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by calling the Disconnect method (or by setting the Action property to ACTION_DISCONNECT).

While handling the OnPostB event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnPostB events.

Example

```
Sub Nntp_OnPostB (Bytes As Long)
  Dim Buffer(1 to Bytes) As Byte
  Dim i As Integer

  If (Siz > Bytes) Then
    For i = 1 to Bytes
      Get #1, , Buffer(i)
    Next i
    Result = Nntp.PostB(Buffer, Bytes)           ' send message
    Siz = Siz - Bytes
  ElseIf (Siz > 0) Then
    For i = 1 to Siz
      Get #1, , Buffer(i)
    Next i
    Result = Nntp.PostB(Buffer, Siz)           ' send message
    Siz = 0
  Else
    Result = Nntp.PostB(Buffer, 0)             ' end of message
  End If
End Sub
```

3.13 OnRetrieve

Summary

More data of article to be retrieved has been received.

Description

The OnRetrieve event occurs in response to the RetrieveAll, RetrieveBody or RetrieveHead method (or to the Action property being set to ACTION_RETRIEVE, after the AccessType property is set). The event delivers the news article from the NNTP server.

If the RetrieveStat method is used (or if the AccessType property is set to ARTICLE_STAT), then the response from the server will **not** be delivered through the OnRetrieve event. The application must check the response by accessing the Reply property.

If the Abort method is used (or if the Action property is set to ACTION_ABORT) during this event, then the retrieve action in progress will be canceled and no further such events will occur. Aborting resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. The application must still close the connection by calling the Disconnect method (or by setting the Action property to ACTION_DISCONNECT).

While handling the OnRetrieve event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnRetrieve events.

Example

```
Sub Nntp_OnRetrieve (Article As String)
    List.Text = List.Text + Article
End Sub
```


4. Methods

4.1 Abort

Summary

Abort current action.

Syntax

Boolean Abort ()

Description

The Abort method aborts any list, help, new, post, forward, or retrieve action in progress. This is usually done during an OnList, OnHelp, OnNew, On Post, OnForward or OnRetrieve event to ensure that no further event will occur. This method resets and closes the connection without properly closing down and should not be called under normal circumstances. Once the connection is reset, the application will get an OnClose event. At this point, the application must still close the connection by setting the Action property to ACTION_DISCONNECT or by calling the Disconnect method.

The Abort method takes no parameters and returns a boolean. If the connection is successfully reset and closed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_ABORT.

Example

```
Result = Nntp.Abort ()  
If Result = False Then  
    MsgBox "Unable to abort", 64, "Sample Program"  
End If
```

4.2 AsyncConnect

Summary

Initiate an asynchronous connect to the NNTP server.

Syntax

Boolean AsyncConnect (*Host*)
Host String

Description

The AsyncConnect method initiates an asynchronous connection to the NNTP server. This method returns immediately, without waiting for the connection to be established. Instead the OnConnect event will occur later once the connection has been established (this may take several seconds).

The AsyncConnect method takes a host name (*Host*) as its parameter and returns a boolean. The *Host* parameter must be the name or internet address (in dotted decimal notation) of the NNTP server. If a connection is initiated successfully, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The Result property is set before the OnError event is fired if the connection could not be established. The value of the Result property can be checked to determine the nature of the error.

Once the connection is established the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

An asynchronous connect can be canceled at any time by calling the CancelAsyncConnect method, or by setting the Action property to ACTION_CANCEL_ASYNC_CONNECT.

Calling the AsyncConnect method is equivalent to setting the Action property to ACTION_ASYNC_CONNECT.

Example

```
Result = Nntp.AsyncConnect ("speedy.distinct.com")
If Result = False Then
    MsgBox "Unable to connect to server", 64, "Sample Program"
End If
```

4.3 CancelAsyncConnect

Summary

Cancel an asynchronous connect to the NNTP server.

Syntax

Boolean CancelAsyncConnect ()

Description

The CancelAsyncConnect method cancels an asynchronous connect to the NNTP server. This method can be called after the AsyncConnect method or after the Action property has been set to ACTION_ASYNC_CONNECT.

This method takes no parameters and returns a boolean. If the asynchronous connect can be canceled successfully, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

An OnError event will be fired if the CancelAsyncConnect operation is unsuccessful. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

Calling the CancelAsyncConnect method is equivalent to setting the Action property to CANCEL_ACTION_ASYNC_CONNECT.

Example

```
Nntp.AsyncConnect("speedy.distinct.com")
Result = Nntp.CancelAsyncConnect ()
If Result = False Then
    MsgBox "Unable to cancel asynchronous connect", 64, "Sample Program"
End If
```

4.4 Connect

Summary

Connect to NNTP server.

Syntax

Boolean Connect (*Host*)
Host String

Description

The Connect method establishes a connection to the NNTP server.

The Connect method takes a host name (*Host*) as its parameter and returns a boolean. The *Host* parameter must be the name or internet address (in dotted decimal notation) of the NNTP server. If a connection is successfully established, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. These events will occur before the next statement (i.e. the statement following the call to the Connect method) is executed. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

Calling this method is equivalent to setting the Action property to ACTION_CONNECT.

Example

```
Result = Nntp.Connect ("speedy.distinct.com")
If Result = False Then
    MsgBox "Unable to connect to server", 64, "Sample Program"
End If
```

4.5 Direct

Summary

Send a command to the NNTP server.

Syntax

Boolean Direct (*Cmd*)
Cmd String

Description

This function sends the command specified by the *Cmd* parameter directly to the NNTP server. The command is sent *as is* to the NNTP server over the previously established connection.

The Direct method takes a command (*Cmd*) to send to the server and returns a boolean. The OnDirect event will be fired to deliver the response returned by the server. If the command was successfully executed by the server then the method returns True; otherwise it returns False. In case of an error the OnError event will be fired.

The application can set the size of the buffer that would receive the reply from the NNTP server in the DirectBufSize property. If the buffer is not large enough to hold the entire reply from the server the NNTP ActiveX control will fire an error (ERR_INSUFFICIENT_BUFSIZE) and the OnDirect method will be fired only with the truncated data. The application at this point can check the DirectBufSize property to find out the required length of the buffer.

The application should ensure that the method was successfully executed by checking the return value. This method also sets the Reply, ReplyCode and Result properties. The value of the Result property can be checked to determine the nature of an error.

Example

```
Nntp.DirectBufSize = 2048
Result = Nntp.Direct ("Help" + Chr(10))
If (Result = False)Then
    MsgBox "Unable to execute direct command", 64, "Sample Program"
End If
```

4.6 Disconnect

Summary

Closes connection to the NNTP server.

Syntax

Boolean Disconnect ()

Description

Once a connection is no longer needed, the session can be terminated by calling the Disconnect method. An application must close all connections it has created before it quits.

The Disconnect method takes no parameters and returns a boolean. If a connection is successfully terminated, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_DISCONNECT.

Example

```
Result = Nntp.Disconnect ()  
If Result = False Then  
    MsgBox "Unable to disconnect", 64, "Sample Program"  
End If
```

4.7 Forward

Summary

Forward an article.

Syntax

Boolean Forward (*ArticleId*)
ArticleId String

Description

The Forward method forwards a news article specified by *ArticleId* to the current server. OnForward events will occur to forward the article.

The Forward method takes an article id (*ArticleId*) as its parameter and returns a boolean. The *ArticleId* must be a valid article id. If the article is successfully forwarded, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_FORWARD.

Example

```
Result = Nntp.Forward (ArticleId)
If Result = False Then
    MsgBox "Unable to forward article", 64, "Sample Program"
End If
```

4.8 ForwardB

Summary

Forward an article with binary data.

Syntax

Boolean ForwardB (*Buffer*, *Bytes*)

Buffer Variant

Bytes Long

Description

The ForwardB method forwards an article with binary data over an established connection. The application should not pass more data than the maximum specified by the Bytes argument of the OnForwardB event to this method.

The ForwardB method takes a buffer (*Buffer*) and the number of bytes to be forwarded (*Bytes*) as its parameters and returns a boolean. The *Buffer* parameter should be an array of bytes data type. The *Bytes* parameter must contain the number of bytes of data to forward. If the data is successfully sent, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Example

```
Result = Nntp.ForwardB (Buffer, Bytes)
```

```
If Result = False Then
```

```
    MsgBox "Cannot forward article with binary data", 64, "Sample Program"
```

```
End If
```

4.9 FwConnect

Summary

Connect to NNTP server via a firewall.

Syntax

Boolean **FwConnect** (*FwServer*, *FwPort*, *Host*)

<i>FwServer</i>	String
<i>FwPort</i>	Integer
<i>Host</i>	String

Description

The FwConnect method establishes a connection to the NNTP server via a firewall.

If the NNTP client machine is located on a different subnet than the remote NNTP server machine and the only form of communication between these two machines is through a firewall gateway, then the built-in firewall support of the NNTP ActiveX control can be used to establish an NNTP session.

The FwConnect method takes a firewall server name (*FwServer*), a firewall port (*FwPort*), and a host name (*Host*) as its parameters and returns a boolean. The *FwServer* parameter must be the name or internet address (in dotted decimal notation) of the firewall server. The *FwPort* parameter must be the firewall service port. The *Host* parameter must be the name or internet address (in dotted decimal notation) of the NNTP server. If the *Host* is the internet address then the *FwAddrType* property must be set to FW_ADDR_IP4 and if it is a machine name then the *FwAddrType* property must be set to FW_ADDR_DNS, by default the *FwAddrType* property has the value FW_ADDR_IP4.

The application can also set the *FwAuthMethods* property if it wants to specify any authentication that needs to be negotiated before an actual connection is established. Only the Username/Password (*FwUsername* and *FwPassword*) authentication is supported currently. An authentication is only possible if the Socks version is 5, version 4 Socks server does not support any authentication protocols. If the Socks version is 4 then the application must provide a valid user id (*FwUsername*).

The version of the socks server can be specified in the *FwSocksVer* property, if the application leaves this property empty then the Distinct firewall control will automatically detect the Socks version.

If a connection is successfully established, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. These events will occur before the next statement (i.e. the statement following the call to the FwConnect method) is executed. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

Calling this method is equivalent to setting the Action property to ACTION_FW_CONNECT.

Example

```
Result = Nntp.FwConnect ("sparky.distinct.com", 1080, "speedy.distinct.com")
If Result = False Then
    MsgBox "Unable to connect to server via a firewall", 64, "Sample Program"
End If
```

4.10 Help

Summary

Show help information.

Syntax

Boolean Help ()

Description

The Help method provides the application with a list of all commands available on the NNTP server with an explanation of each command. The information will be delivered through OnHelp events.

The Help method takes no parameters and returns a boolean. If the Help command is successfully executed on the server, then the method will return True; otherwise, it will return False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_HELP.

Example

```
Result = Nntp.Help ()  
If Result = False Then  
    MsgBox "Unable to obtain helpful information", 64, "Sample Program"  
End If
```

4.11 List

Summary

List all available news groups.

Syntax

Boolean List ()

Description

The List method lists all available news groups on the server. The information will be delivered through OnList events.

The List method takes no parameters and returns a boolean. If a list of available news groups can be successfully obtained, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_LIST.

Example

```
Result = Nntp.List ()  
If Result = False Then  
    MsgBox "Unable to obtain list of available news groups", 64, "Sample Program"  
End If
```

4.12 NewArticles

Summary

Get new news articles Id.

Syntax

Boolean NewArticles (*CmdLine*)
CmdLine String

Description

The NewArticles method retrieves all the article Ids for the articles, in the news group that is currently specified by the Group property, created since the date and time specified by the *CmdLine* parameter. The list of news articles will be delivered through one or more OnNew events.

The NewArticles method takes (*CmdLine*) the date in the form of YYMMDD HHMMSS as its parameter and returns a boolean. Note that HH represents the hour on a 24 hour clock; MM and SS are 00-59. If an empty string is passed as the *CmdLine* parameter, then an error is returned. If the articleId for all the articles are successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_NEW with the NewType property set to NEW_NEWS.

Example

```
Result = Nntp.NewArticles ("990101 000000")
If Result = False Then
    MsgBox "Unable to list news articles", 64, "Sample Program"
End If
```

4.13 NewGroups

Summary

Get new news groups.

Syntax

Boolean NewGroups (*CmdLine*)
CmdLine String

Description

The NewGroups method retrieves all the new groups created since the date and time specified by the new command line. The list of news groups will be delivered through one or more OnNew events.

The NewGroups method takes a command line (*CmdLine*) as its parameter and returns a boolean. If an empty string is passed as the *CmdLine* parameter, then an error is returned. Otherwise, all new groups created since the specified date and time are retrieved. If the new groups are successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_NEW with the NewType property set to NEW_GROUPS.

Example

```
Result = Nntp.NewGroups ("960101 000000")  
If Result = False Then  
    MsgBox "Unable to retrieve news groups", 64, "Sample Program"  
End If
```

4.14 Next

Summary

Go to next article.

Syntax

Boolean Next ()

Description

The Next method sets the current news article in the selected news group to the next article. The Reply property can be accessed to check the response from the server for error returns. The ReplyCode property can also be used by the application to check the severity of an error.

The Next method takes no parameters and returns a boolean. If the next article can successfully be set as the current news article, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_POSITION with the Position property set to POSITION_NEXT.

Example

```
Result = Nntp.Next ()
If Result = False Then
    MsgBox "Unable to set next article as current article", 64, "Sample Program"
    Exit Sub
End If
```

4.15 Overview

Summary

Retrieve an overview of a specified range of news articles.

Syntax

Boolean Overview (StartArticleNo, EndArticleNo)

StartArticleNo	Long
EndArticleNo	Long

Description

The Overview method retrieves an overview of a specified range of news articles. The news articles summaries will be delivered through one or more OnOverview events.

The Overview method takes a start article number (StartArticleNo) and an end article number (EndArticleNo) as its parameters and returns a boolean. If the summaries of the specified range of news articles can successfully be retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_OVERVIEW.

Example

```
Result = Nntp.Overview (1234, 1240)
If Result = False Then
    MsgBox "Unable to retrieve news articles summaries", 64, "Sample Program"
    Exit Sub
End If
```

4.16 Post

Summary

Post an article to a news group.

Syntax

Boolean Post ()

Description

The Post method posts an article to a news group. OnPost events will occur to receive the article header and the body. The required headers, such as the date and the news group, must be specified.

The Post method takes no parameters and returns a boolean. If an article can successfully be posted to a news group, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_POST.

Example

```
Result = Nntp.Post ()  
If Result = False Then  
    MsgBox "Unable to post article", 64, "Sample Program"  
    Exit Sub  
End If
```

4.17 PostB

Summary

Post an article with binary data.

Syntax

Boolean PostB (*Buffer*, *Bytes*)

Buffer Variant

Bytes Long

Description

The PostB method forwards an article with binary data over an established connection. The application should not pass more data than the maximum specified by the Bytes argument of the OnPostB event to this method.

The PostB method takes a buffer (*Buffer*) and the number of bytes to be posted (*Bytes*) as its parameters and returns a boolean. The *Buffer* parameter should be an array of bytes data type. The *Bytes* parameter must contain the number of bytes of data to post. If the data is successfully sent, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Example

```
Result = Nntp.PostB (Buffer, Bytes)
```

```
If Result = False Then
```

```
    MsgBox "Cannot post article with binary data", 64, "Sample Program"
```

```
End If
```

4.18 PostBuf

Summary

Post an article to a news group.

Syntax

Boolean PostBuf (*Buffer*, *Bytes*)
Buffer Variant Far*
Bytes Long

Description

The PostBuf method posts an article to a news group. The PostBuf method can be called several times to post the article header and the body. The required headers, such as the date and the news group, must be specified.

The PostBuf method takes a pointer to a Variant (*Buffer*) containing data and the number of bytes to be posted (*Bytes*) as its parameters and returns a boolean. The *Buffer* parameter should be a pointer to the Variant data type. The *Bytes* parameter must contain the number of bytes of data to post. If the data is successfully sent, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. One last call should be made with the *Bytes* parameter set to 0 and the *Buffer* pointing to an empty string to signify the end of data. The Variant data type should be either of type VT_ARRAY or of type VT_BSTR, if the data type is anything else then the method will return FALSE.

Example

```
Sub OnSend ()
Dim Buffer As Variant
  For i = 0 to 5
    If (i = 0) Then
      Buffer = "Date: " + Nntp.Date + Chr(10)           ' date
    ElseIf (i = 1) Then
      Buffer = "From: John" + Chr(10)                 ' sender
    ElseIf (i = 2) Then
      Buffer = "NewsGroups: misc.test" + Chr(10)      ' news groups
    ElseIf (i = 3) Then
      Buffer = "Subject: Testing" + Chr(10)           ' subject
    ElseIf (i = 4)
      Buffer = Chr(10) + "This is a test"             ' data
    Else
      Buffer = ""                                     ' end of data
    End If
    Result = Nntp.PostBuf (Buffer, Len (Buffer))
    If (Result = FALSE) Then
      MsgBox "Unable Post article", 64, "Sample Program"
    Exit Sub
  End If
Next i
End Sub
```

4.19 Previous

Summary

Go to previous article.

Syntax

Boolean Previous ()

Description

The Previous method sets the current news article in the selected news group to the previous article. The Reply property can be accessed to check the response from the server for error returns. The ReplyCode property can also be used by the application to check the severity of an error.

The Previous method takes no parameters and returns a boolean. If the previous article can be successfully set as the current news article, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_POSITION with the Position property to POSITION_PREV.

Example

```
Result = Nntp.Previous ()
If Result = False Then
    MsgBox "Unable to set previous article as current article", 64, "Sample Program"
End If
```

4.20 RetrieveAll

Summary

Retrieve an entire article.

Syntax

Boolean RetrieveAll (*ArticleId*)
ArticleId String

Description

The RetrieveAll method retrieves the entire specified article for reading. OnRetrieve events will deliver the entire news article. The application must access the Reply property to obtain the response from the server.

The RetrieveAll method takes an article id (*ArticleId*) as its parameter and returns a boolean. If an empty string is passed as the *ArticleId* parameter, then the entire current article is retrieved. If the entire news article is successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_RETRIEVE with the AccessType property set to ARTICLE_ALL.

Example

```
ArticleId = "123456"  
Result = Nntp.RetrieveAll (ArticleId)  
If Result = False Then  
    MsgBox "Unable to retrieve the entire article", 64, "Sample Program"  
End If
```

4.21 RetrieveBody

Summary

Retrieve the message portion of an article.

Syntax

Boolean RetrieveBody (*ArticleId*)
ArticleId String

Description

The RetrieveBody method retrieves the message portion of an article for reading. OnRetrieve events will deliver the message portion of the news article. The application must access the Reply property to obtain the response from the server.

The RetrieveBody method takes an article id (*ArticleId*) as its parameter and returns a boolean. If an empty string is passed as the *ArticleId* parameter, then the message portion of the current article is retrieved. If the message portion of the news article is successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_RETRIEVE with the AccessType property set to ARTICLE_BODY.

Example

```
Result = Nntp.RetrieveBody ("")
If Result = False Then
    MsgBox "Unable to retrieve the message portion of article", 64, "Sample Program"
End If
```

4.22 RetrieveHead

Summary

Retrieve the article header.

Syntax

Boolean RetrieveHead (*ArticleId*)
ArticleId String

Description

The RetrieveHead method retrieves the header of the specified article for reading. OnRetrieve events will deliver the header of the news article. The application must access the Reply property to obtain the response from the server.

The RetrieveHead method takes an article id (*ArticleId*) as its parameter and returns a boolean. If an empty string is passed as the *ArticleId* parameter, then the header of the current article is retrieved. If the header of the news article is successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_RETRIEVE with the AccessType property set to ARTICLE_HEAD.

Example

```
Result = Nntp.RetrieveHead ("")  
If Result = False Then  
    MsgBox "Unable to retrieve the header of article", 64, "Sample Program"  
End If
```

4.23 RetrieveStat

Summary

Retrieve article statistics.

Syntax

Boolean RetrieveStat (*ArticleId*)
ArticleId String

Description

The RetrieveStat method retrieves the statistics of the specified article for viewing. The article statistics will **not** be delivered via OnRetrieve events. The application must access the Reply property to obtain the response from the server.

The RetrieveStat method takes an article id (*ArticleId*) as its parameter and returns a boolean. If an empty string is passed as the *ArticleId* parameter, then the statistics of the current article is retrieved. If the news article statistics is successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_RETRIEVE with the AccessType property set to ARTICLE_STAT.

Example

```
Result = Nntp.RetrieveStat ("")
If Result = False Then
    MsgBox "Unable to retrieve the article statistics", 64, "Sample Program"
End If
```

4.24 Set

Summary

Go to a specified article.

Syntax

Boolean Set (*ArticleId*)
ArticleId String

Description

The Set method sets the current news article in the selected news group to the specified article. The Reply property can be accessed to check the response from the server for error returns. The ReplyCode property can also be used by the application to check the severity of an error.

The Set method takes an article id (*ArticleId*) as its parameter and returns a boolean. The *ArticleId* has to be a valid article number. If the specified article can be successfully set as the current news article, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_POSITION with the Position property set to POSITION_SET.

Example

```
ArticleId = "98615"  
Result = Nntp.Set (ArticleId)  
If Result = False Then  
    MsgBox "Unable to set article as current", 64, "Sample Program"  
End If
```

4.25 Slave

Summary

Expand list of mail addresses.

Syntax

Boolean Slave ()

Description

The Slave method informs the current NNTP server that the local machine is acting as an intermediate or "slave" NNTP server. This information is only sent to advise the NNTP server and may be ignored by the server. It is normally used only by NNTP servers.

The Slave method takes no parameters and returns a boolean. If the information can be successfully sent to the server, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value. This method also sets the Result property. The value of the Result property can be checked to determine the nature of the error.

Calling this method is equivalent to setting the Action property to ACTION_SLAVE.

Example

```
Result = Nntp.Slave ()  
If Result = False Then  
    MsgBox "Unable to execute command", 64, "Sample Program"  
End If
```

